THAT WHICH IS CLAIMED IS:

1. A method of controlling the generation of interrupt command in a microprocessor system, comprising

associating pre-established priority values to respective possible interrupts (INT0, ..., INTm),

storing received interrupts (INT0, ..., INTm), in a plurality of pending interrupt registers (INT PENDING REG) and storing the priority values associated with the interrupts in a plurality of priority registers (PRIORITY REGISTERS),

if no interrupt service routine (ISR) is running, carrying out the following operations:

comparing the stored priority values of the pending interrupts identifying which of them (INTn) has the highest priority,

serving the highest priority interrupt by generating an interrupt command (nIRQ) and an interrupt vector (IRQ VECTOR) identifying a respective interrupt service routine (ISR) to be run,

canceling the served interrupt and its priority value from said registers,

characterized in that the method comprises the steps of

connecting a plurality of counters (PRIORITY COUNTERS) in cascade of said plurality of pending interrupts registers (INT PENDING REG);

for each interrupt received, loading in a respective counter of said plurality the corresponding pre-established priority value associated to the received interrupt;

incrementing at pre-established intervals the

value loaded in said counters;

comparing the updated priority values contained in said periodically incremented counters for identifying the interrupt (INTn) of the highest priority.

2. The method of claim 1, wherein said incrementing is done by increment of signals of different period depending on the interrupt that is associated to the respective counter of said plurality of counters.

3. The method of claim 2, wherein the increment signal of a certain counter has a period shorter than the ratio between an allowed maximum latency of the pertinent interrupt and the difference between the maximum and minimum priority values.

4. The method of claim 1, further comprising for managing nested interrupt

storing in a memory buffer (CURR IRQ PRIORITY REG) the priority of an interrupt being served;

identifying the interrupt (INTn) of highest priority by comparing the priority values stored in the counters respectively associated to the stored interrupts and in said memory buffer (CURR IRQ PRIORITY REG);

should an incoming new interrupt have a higher priority than the interrupt being served, stopping the servicing in progress of the interrupt and storing in a stack register (INT PRIORITY STACK) the priority of the interrupt whose servicing was suspended;

serving the interrupt corresponding to the highest priority value stored in said stack register when said priority results to be the highest among the priorities of all pending interrupts, and canceling from said stack register (INT PRIORITY STACK) the priority value of the corresponding served interrupt.

5. A control circuit for generating interrupt commands in a microprocessor system, comprising

a pending interrupts register (INT PENDING REG), storing interrupts (INT0, ..., INTm) received from peripherals,

a plurality of priority registers (PRIORITY REGISTERS) storing priority values associated to respective interrupts (INT0, ..., INTm),

a priority comparing circuit (IRQ MASK AND PRIORITY LOGIC) coupled to said priority registers (PRIORITY REGISTERS) and to said pending interrupts register (INT PENDING REG), generating an interrupt request signal (IRQ REQ) and an internal signal (HIGHEST PRIORITY INT) representing the priority of the interrupt (INTn) of highest priority stored in said pending interrupts register (INT PENDING REG),

a logic processing circuit (IRQ SM, IRQ VECTOR REG, IRQ VECTOR) of said interrupt request signal (IRQ REQ) and of said internal signal (HIGHEST PRIORITY INT) for sending to a system microprocessor an interrupt command (nIRQ) and an interrupt vector (IRQ VECTOR) identifying a certain interrupt service routine (ISR) to be executed,

characterized by comprising further

a plurality of counters (PRIORITY COUNTERS)

coupled in cascade to said plurality of priority registers (PRIORITY REGISTERS) and means for initializing any of the counters with the priority value of a certain interrupt (INT0, ..., INTm) to be served and for periodically incrementing the value by a respective increment signal (PRIORITY TRIGGERS) input to the register;

said priority comparing circuit (IRQ MASK AND PRIORITY LOGIC) reading from said counters (PRIORITY COUNTERS) the priority associated to each interrupt (INT0, ..., INTm) stored in said pending interrupts register (INT PENDING REG).

6. The control circuit of claim 5, wherein said increment signals (PRIORITY TRIGGERS) are derived from a system clock signal or from externally generated timing signals fed through dedicated pins of the control circuit.

7. The control circuit of claim 5, wherein the period of said increment signals (PRIORITY TRIGGERS) is a multiple of the period of said system clock signal.

8. The control circuit of claim 5, wherein said counters (PRIORITY COUNTERS) receive respective increment signals of different period.

9. The control circuit of claim 5 adapted to manage nested interrupts, wherein said logic processing circuit comprises further

a memory buffer (CURR IRQ PRIORITY REG) coupled to said priority comparing circuit (IRQ MASK

AND PRIORITY LOGIC) for storing the priority of an interrupt being served represented by said internal signal (HIGHEST PRIORITY INT);

a stack register (INT PRIORITY STACK) coupled to said memory buffer (CURR IRQ PRIORITY REG) storing the priority of interrupts whose servicing was suspended because of the incoming of an interrupt of higher priority.